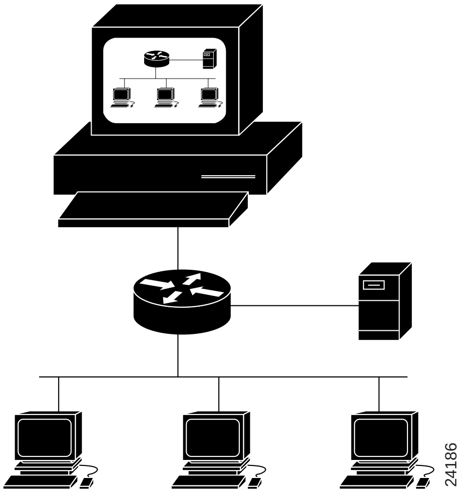
Simple Network Management Protocol (SNMP)

Background

The Simple Network Management Protocol(SNMP)is an application-layer protocol that facilitates the exchange of management information between network devices. It is part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth.

Two versions of SNMP exist: SNMP Version 1 (SNMPv1) and SNMP Version 2 (SNMPv2). Both versions have a number of features in common, but SNMPv2 offers enhancements, such as additional protocol operations. Standardization of yet another version of SNMP—SNMP Version 3 (SNMPv3)—is pending. This chapter provides descriptions of the SNMPv1 and SNMPv2 protocol operations. Figure 52-1 illustrates a basic network managed by SNMP.

Figure 52-1 SNMP facilitates the exchange of network information between devices.



SNMP Basic Components

An SNMP managed network consists of three key components: *managed devices*, *agents*, and *network-management systems* (NMSs).

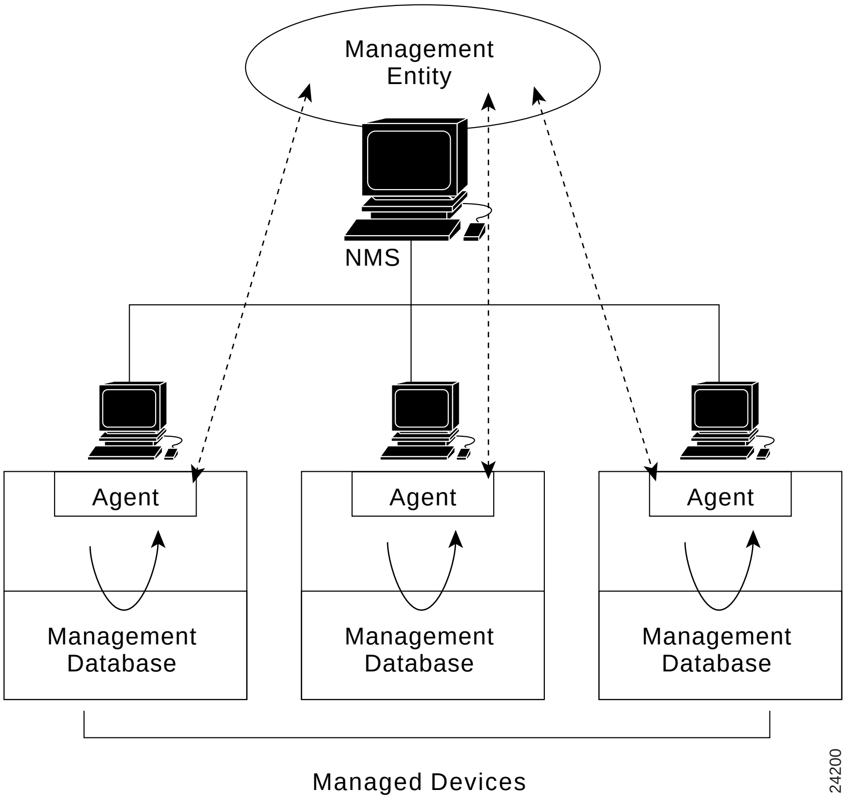
A managed device is a network node that contains an SNMP agent and resides on a managed network. Managed devices collect and store management information and make this information available to NMSs using SNMP. Managed devices, sometimes called network elements, can be routers and access servers, switches and bridges, hubs, computer hosts, or printers.

An agent is a network-management software module that resides in a managed device. An agent has local knowledge of management information and translates that information into a form compatible with SNMP.

An NMS executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. One or more NMSs must exist on any managed network.

Figure 52-2 illustrates the relationship between these three components.

Figure 52-2 An SNMP managed network consists of managed devices, agents, and NMSs.



SNMP Basic Commands

Managed devices are monitored and controlled using four basic SNMP commands: *read*, *write*, *trap*, and *traversal operations*.

- The read command is used by an NMS to monitor managed devices. The NMS examines different variables that are maintained by managed devices.
- The write command is used by an NMS to control managed devices. The NMS changes the values of variables stored within managed devices.
- The trap command is used by managed devices to asynchronously report events to the NMS. When certain types of events occur, a managed device sends a trap to the NMS.

Traversal operations are used by the NMS to determine which variables a managed device supports and to sequentially gather information in variable tables, such as a routing table.

SNMP Management Information Base (MIB)

A *Management Information Base (MIB)* is a collection of information that is organized hierarchically. MIBs are accessed using a network-management protocol such as SNMP. They are comprised of managed objects and are identified by object identifiers.

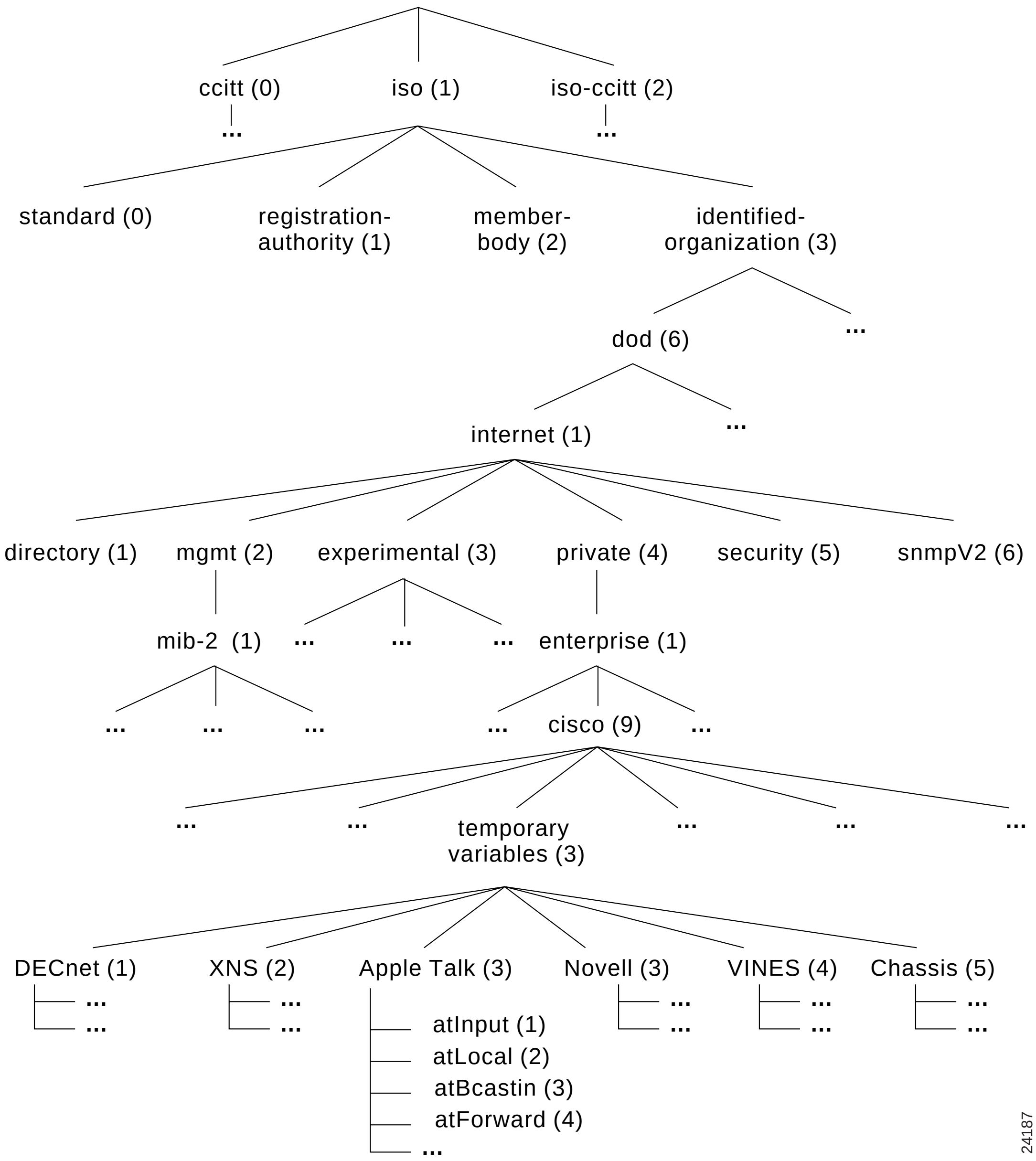
A managed object (sometimes called a MIB object, an object, or a MIB) is one of any number of specific characteristics of a managed device. Managed objects are comprised of one or more object instances, which are essentially variables.

Two types of managed objects exist: *scalar* and *tabular.* Scalar objects define a single object instance. Tabular objects define multiple related object instances that are grouped together in MIB tables.

An example of a managed object is *at Input*, which is a scalar object that contains a single object instance, the integer value that indicates the total number of input AppleTalk packets on a router interface.

An object identifier (or object ID) uniquely identifies a managed object in the MIB hierarchy. The MIB hierarchy can be depicted as a tree with a nameless root, the levels of which are assigned by different organizations. Figure 52-3 illustrates the MIB tree.

Figure 52-3 The MIB tree illustrates the various hierarchies assigned by different organizations.



The top-level MIB object IDs belong to different standards organizations, while lower-level object IDs are allocated by associated organizations.

Vendors can define private branches that include managed objects for their own products. MIBs that have not been standardized typically are positioned in the experimental branch.

The managed object *at Input* can be uniquely identified either by the object name—*iso.identified-organization.dod.internet.private.enterprise.cisco.temporary variables.AppleTalk.atInput*—or by the equivalent object descriptor: 1.3.6.1.4.1.9.3.3.1.

SNMP and Data Representation

SNMP must account for and adjust to incompatibilities between managed devices. Different computers use different data-representation techniques, which can compromise the ability of SNMP to exchange information between managed devices. SNMP uses a subset of Abstract Syntax Notation One (ASN.1) to accommodate communication between diverse systems.

SNMP Version 1 (SNMPv1)

SNMP Version 1 (SNMPv1) is the initial implementation of the SNMP protocol. It is described in Request For Comments (RFC) 1157 and functions within the specifications of the Structure of Management Information (SMI). SNMPv1 operates over protocols such as User Datagram Protocol (UDP), Internet Protocol (IP), OSI Connectionless Network Service (CLNS), AppleTalk Datagram-Delivery Protocol (DDP), and Novell Internet Packet Exchange (IPX). SNMPv1 is widely used and is the de facto network-management protocol in the Internet community.

SNMPv1 and Structure of Management Information (SMI)

The Structure of Management Information (SMI) defines the rules for describing management information, using Abstract Syntax Notation One (ASN.1). The SNMPv1 SMI is defined in Request For Comments (RFC) 1155. The SMI makes three key specifications: ASN.1 data types, SMI-specific data types, and SNMP MIB tables.

SNMPv1 and ASN1 Data Types

The SNMPv1 SMI specifies that all managed objects have a certain subset of Abstract Syntax Notation One (ASN.1) data types associated with them. Three ASN.1 data types are required: *name*, *syntax*, and *encoding*. The name serves as the object identifier (object ID). The syntax defines the data type of the object (for example, integer or string). The SMI uses a subset of the ASN.1 syntax definitions. The encoding data describes how information associated with a managed object is formatted as a series of data items for transmission over the network.

SNMPv1 and SMI-Specific Data Types

The SNMPv1 SMI specifies the use of a number of SMI-specific data types, which are divided into two categories: *simple data types* and *application-wide data types*.

Three simple data types are defined in the SNMPv1 SMI, all of which are unique values: *integers*, *octet strings*, and *object IDs*. The integer data type is a signed integer in the range of -2,147,483,648 to 2,147,483,647. Octet strings are ordered sequences of zero to 65,535 octets. Object IDs come from the set of all object identifiers allocated according to the rules specified in ASN.1.

Seven application-wide data types exist in the SNMPv1 SMI: *network addresses*, *counters*, *gauges*, *time ticks*, *opaques*, *integers*, and *unsigned integers*. Network addresses represent an address from a particular protocol family. SNMPv1 supports only 32-bit IP addresses. Counters are nonnegative integers that increase until they reach a maximum value and then return to zero. In SNMPv1, a 32-bit counter size is specified. Gauges are nonnegative integers that can increase or decrease but retain the maximum value reached. A time tick represents a hundredth of a second since some event. An opaque represents an arbitrary encoding that is used to pass arbitrary information strings that do not conform to the strict data typing used by the SMI. An integer represents signed integer-valued information. This data type redefines the integer data type, which has arbitrary precision in ASN.1

but bounded precision in the SMI. An unsigned integer represents unsigned integer-valued information and is useful when values are always nonnegative. This data type redefines the integer data type, which has arbitrary precision in ASN.1 but bounded precision in the SMI.

SNMP MIB Tables

The SNMPv1 SMI defines highly structured tables that are used to group the instances of a tabular object (that is, an object that contains multiple variables). Tables are composed of zero or more rows, which are indexed in a way that allows SNMP to retrieve or alter an entire row with a single Get, GetNext, or Set command.

SNMPv1 Protocol Operations

SNMP is a simple request-response protocol. The network-management system issues a request, and managed devices return responses. This behavior is implemented by using one of four protocol operations: *Get*, *GetNext*, *Set*, and *Trap*. The Get operation is used by the NMS to retrieve the value of one or more object instances from an agent. If the agent responding to the Get operation cannot provide values for all the object instances in a list, it does not provide any values. The GetNext operation is used by the NMS to retrieve the value of the next object instance in a table or list within an agent. The Set operation is used by the NMS to set the values of object instances within an agent. The Trap operation is used by agents to asynchronously inform the NMS of a significant event.

SNMP Version 2 (SNMPv2)

SNMP Version 2 (SNMPv2) is an evolution of the initial version, SNMPv1. Originally, SNMPv2 was published as a set of proposed Internet standards in 1993; currently, it is a Draft Standard. As with SNMPv1, SNMPv2 functions within the specifications of the Structure of Management Information (SMI). In theory, SNMPv2 offers a number of improvements to SNMPv1, including additional protocol operations.

SNMPv2 and Structure of Management Information (SMI)

The Structure of Management Information (SMI) defines the rules for describing management information, using Abstract Syntax Notation One (ASN.1).

The SNMPv2 SMI is described in RFC 1902. It makes certain additions and enhancements to the SNMPv1 SMI-specific data types, such as including *bit strings*, *network addresses*, and *counters*. Bit strings are defined only in SNMPv2 and comprise zero or more named bits that specify a value. Network addresses represent an address from a particular protocol family. SNMPv1 supports only 32-bit IP addresses, but SNMPv2 can support other types of addresses as well. Counters are non-negative integers that increase until they reach a maximum value and then return to zero. In SNMPv1, a 32-bit counter size is specified. In SNMPv2, 32-bit and 64-bit counters are defined.

SMI Information Modules

The SNMPv2 SMI also specifies information modules, which specify a group of related definitions. Three types of SMI information modules exist: *MIB modules*, *compliance statements*, and *capability statements*. MIB modules contain definitions of interrelated managed objects. Compliance statements provide a systematic way to describe a group of managed objects that must be implemented for conformance to a standard. Capability statements are used to indicate the precise level of support that an agent claims with respect to a MIB group. An NMS can adjust its behavior toward agents according to the capabilities statements associated with each agent.

SNMPv2 Protocol Operations

The Get, GetNext, and Set operations used in SNMPv1 are exactly the same as those used in SNMPv2. SNMPv2, however, adds and enhances some protocol operations. The SNMPv2 Trap operation, for example, serves the same function as that used in SNMPv1. It, however, uses a different message format and is designed to replace the SNMPv1 Trap.

SNMPv2 also defines two new protocol operations: *GetBulk* and *Inform*. The GetBulk operation is used by the NMS to efficiently retrieve large blocks of data, such as multiple rows in a table. GetBulk fills a response message with as much of the requested data as will fit. The Inform operation allows one NMS to send trap information to another NMS and receive a response. In SNMPv2, if the agent responding to GetBulk operations cannot provide values for all the variables in a list, it provides partial results.

SNMP Management

SNMP is a distributed-management protocol. A system can operate exclusively as either an NMS or an agent, or it can perform the functions of both. When a system operates as both an NMS and an agent, another NMS might require that the system query managed devices and provide a summary of the information learned, or that it report locally stored management information.

SNMP Security

SNMP lacks any authentication capabilities, which results in vulnerability to a variety of security threats. These include *masquerading*, *modification of information*, *message sequence and timing modifications*, and *disclosure*. Masquerading consists of an unauthorized entity attempting to perform management operations by assuming the identity of an authorized management entity. Modification of information involves an unauthorized entity attempting to alter a message generated by an authorized entity so that the message results in unauthorized accounting management or configuration management operations. Message sequence and timing modifications occur when an unauthorized entity reorders, delays, or copies and later replays a message generated by an authorized entity. Disclosure results when an unauthorized entity extracts values stored in managed objects, or learns of notifiable events by monitoring exchanges between managers and agents. Because SNMP does not implement authentication, many vendors do not implement Set operations, thereby reducing SNMP to a monitoring facility.

SNMP Interoperability

As presently specified, SNMPv2 is incompatible with SNMPv1 in two key areas: message formats and protocol operations. SNMPv2 messages use different header and protocol data-unit (PDU) formats than SNMPv1 messages. SNMPv2 also uses two protocol operations that are not specified in SNMPv1. Furthermore, RFC 1908 defines two possible SNMPv1/v2 coexistence strategies: proxy agents and “bilingual” network-management systems.

Proxy Agents

An SNMPv2 agent can act as a proxy agent on behalf of SNMPv1 managed devices, as follows:

- An SNMPv2 NMS issues a command intended for an SNMPv1 agent.
- The NMS sends the SNMP message to the SNMPv2 proxy agent.
- The proxy agent forwards Get, GetNext, and Set messages to the SNMPv1 agent unchanged.

- GetBulk messages are converted by the proxy agent to GetNext messages and then are forwarded to the SNMPv1 agent.
- The proxy agent maps SNMPv1 trap messages to SNMPv2 trap messages and then forwards them to the NMS.

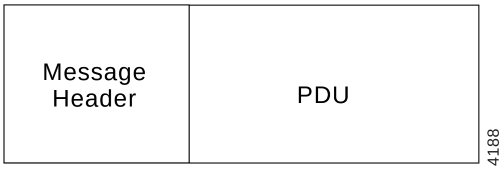
Bilingual Network-Management System

Bilingual SNMPv2 network-management systems support both SNMPv1 and SNMPv2. To support this dual-management environment, a management application in the bilingual NMS must contact an agent. The NMS then examines information stored in a local database to determine whether the agent supports SNMPv1 or SNMPv2. Based on the information in the database, the NMS communicates with the agent using the appropriate version of SNMP.

SNMP Reference: SNMPv1 Message Formats

SNMPv1 messages contain two parts: a message header and a protocol data unit. Figure 52-4 illustrates the basic format of an SNMPv1 message.

Figure 52-4 An SNVPv1 message consists of a header and a PDU.



SNMPv1 Message Header

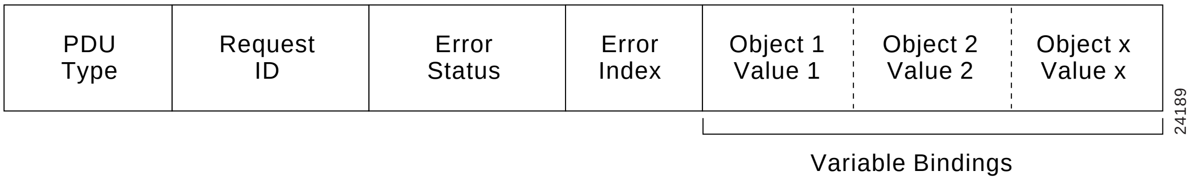
SNMPv1 message headers contain two fields: *Version Number* and *Community Name*. The following descriptions summarize these fields:

- *Version Number*—Specifies the version of SNMP used.
- *Community Name*—Defines an access environment for a group of NMSs. NMSs within the community are said to exist within the same administrative domain. Community names serve as a weak form of authentication because devices that do not know the proper community name are precluded from SNMP operations.

SNMPv1 Protocol Data Unit (PDU)

SNMPv1 PDUs contain a specific command (Get, Set, and so on) and operands that indicate the object instances involved in the transaction. SNMPv1 PDU fields are variable in length, as prescribed by Abstract Syntax Notation One (ASN.1). Figure 52-5 illustrates the fields of the SNMPv1 Get, GetNext, Response, and Set PDUs transactions.

Figure 52-5 SNMPv1 Get, GetNext, Response, and Set PDUs contain the same fields.



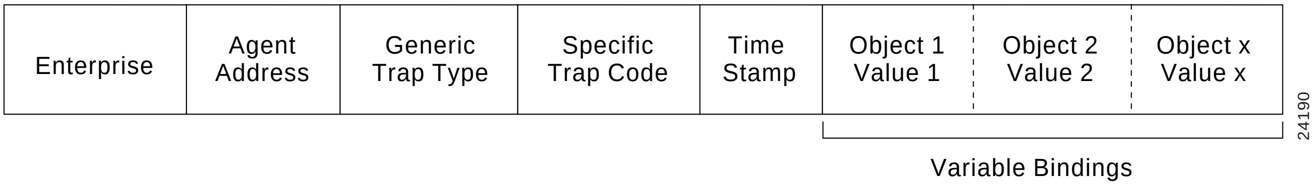
The following descriptions summarize the fields illustrated in Figure 52-5:

- *PDU Type*—Specifies the type of PDU transmitted.
- *Request ID*—Associates SNMP requests with responses.
- *Error Status*—Indicates one of a number of errors and error types. Only the response operation sets this field. Other operations set this field to zero.
- *Error Index*—Associates an error with a particular object instance. Only the response operation sets this field. Other operations set this field to zero.
- *Variable Bindings*—Serves as the data field of the SNMPv1 PDU. Each variable binding associates a particular object instance with its current value (with the exception of Get and GetNext requests, for which the value is ignored).

Trap PDU Format

Figure 52-6 illustrates the fields of the SNMPv1 Trap PDU.

Figure 52-6 The SNMPv1 Trap PDU consists of eight fields.



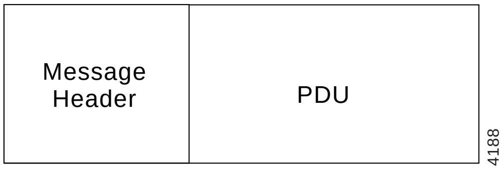
The following descriptions summarize the fields illustrated in Figure 52-6:

- *Enterprise*—Identifies the type of managed object generating the trap.
- *Agent Address*—Provides the address of the managed object generating the trap.
- *Generic Trap Type*—Indicates one of a number of generic trap types.
- *Specific Trap Code*—Indicates one of a number of specific trap codes.
- *Time Stamp*—Provides the amount of time that has elapsed between the last network reinitialization and generation of the trap.
- *Variable Bindings*—The data field of the SNMPv1 Trap PDU. Each variable binding associates a particular object instance with its current value.

SNMP Reference: SNMPv2 Message Format

SNMPv2 messages consist of a header and a PDU. Figure 52-7 illustrates the basic format of an SNMPv2 message.

Figure 52-7 SNMPv2 messages also consist of a header and a PDU.



SNMPv2 Message Header

SNMPv2 message headers contain two fields: *Version Number* and *Community Name*. The following descriptions summarize these fields:

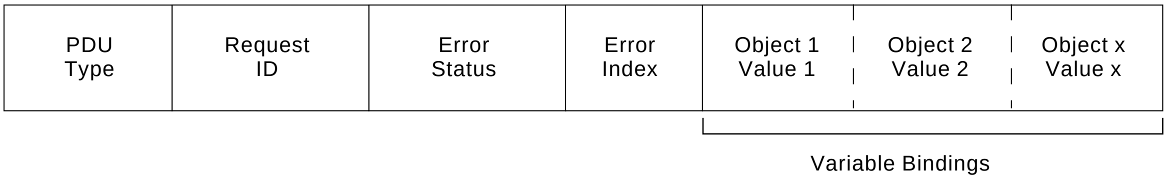
- *Version Number*—Specifies the version of SNMP that is being used.
- *Community Name*—Defines an access environment for a group of NMSs. NMSs within the community are said to exist within the same administrative domain. Community names serve as a weak form of authentication because devices that do not know the proper community name are precluded from SNMP operations.

SNMPv2 Protocol Data Unit (PDU)

SNMPv2 specifies two PDU formats, depending on the SNMP protocol operation. SNMPv2 PDU fields are variable in length, as prescribed by Abstract Syntax Notation One (ASN.1).

Figure 52-8 illustrates the fields of the SNMPv2 Get, GetNext, Inform, Response, Set, and Trap PDUs.

Figure 52-8 SNMPv2 Get, GetNext, Inform, Response, Set, and Trap PDUs contain the same fields.



The following descriptions summarize the fields illustrated in Figure 52-8:

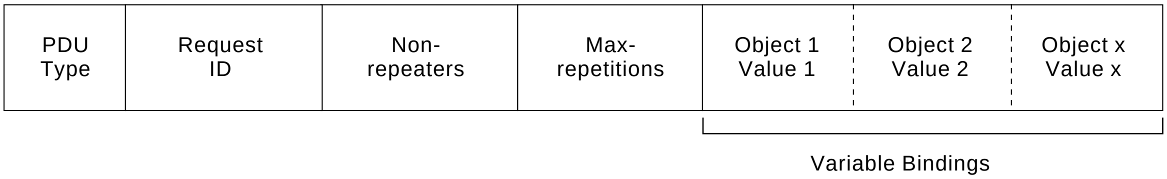
- *PDU Type*—Identifies the type of PDU transmitted (Get, GetNext, Inform, Response, Set, or Trap).
- *Request ID*—Associates SNMP requests with responses.
- *Error Status*—Indicates one of a number of errors and error types. Only the response operation sets this field. Other operations set this field to zero.
- *Error Index*—Associates an error with a particular object instance. Only the response operation sets this field. Other operations set this field to zero.

- *Variable Bindings*—Serves as the data field of the SNMPv2 PDU. Each variable binding associates a particular object instance with its current value (with the exception of Get and GetNext requests, for which the value is ignored).

GetBulk PDU Format

Figure 52-9 illustrates the fields of the SNMPv2 GetBulk PDU.

Figure 52-9 The SNMPv2 GetBulk PDU consists of seven fields.



The following descriptions summarize the fields illustrated in Figure 52-9:

- *PDU Type*—Identifies the PDU as a GetBulk operation.
- *Request ID*—Associates SNMP requests with responses.
- *Non-repeaters*—Specifies the number of object instances in the variable bindings field that should be retrieved no more than once from the beginning of the request. This field is used when some of the instances are scalar objects with only one variable.
- *Max-repetitions*—Defines the maximum number of times that other variables beyond those specified by the non-repeaters field should be retrieved.
- *Variable Bindings*—Serves as the data field of the SNMPv2 PDU. Each variable binding associates a particular object instance with its current value (with the exception of Get and GetNext requests, for which the value is ignored).